# Predicting IPO Underpricing with Genetic Algorithms

**Cristóbal Luque**[1], **David Quintana**[2] and **Pedro Isasi**[3]

[1]Department of Computer Science, Universidad Carlos III de Madrid,
Av. de la Universidad 30, Leganés 28911 (Spain);
Email : cluqueac@gmail.com

[2]Department of Computer Science, Universidad Carlos III de Madrid,
Av. de la Universidad 30, Leganés 28911 (Spain);
Email : dquintan@inf.uc3m.es

[3]Department of Computer Science, Universidad Carlos III de Madrid,
Av. de la Universidad 30, Leganés 28911 (Spain);
Email : isasi@ia.uc3m.es

**ABSTRACT**

*This paper introduces a rule system to predict first-day returns of initial public offerings based on the structure of the offerings. The solution is based on a genetic algorithm using a Michigan approach. The performance of the system is assessed comparing it to a set of widely used machine learning algorithms. The results suggest that this approach offers significant advantages on two fronts: predictive performance and robustness to outlier patterns. The importance of the latter should be emphasized as the results in this domain are very sensitive to their presence.*

**Keywords:** Genetic algorithm, IPO, underpricing.
**Mathematics Subject Classification:** 68T20
**Computing Classification System:** I.2.6, J.4

## 1. INTRODUCTION

Academic literature has documented for a long time the existence of abnormal first-day trading returns on initial public offerings (IPOs). Historically, there has been a substantial difference between the price at which the stock is sold to investors and the closing at the end of the first trading day. The size of gap between the price set by the sellers and their advisors and the market price, once the shares of the company have been trading freely, varies with time periods and industries and is difficult to justify from current financial theory. Ritter and Welch (2002) report an average initial return of 18.8% on a sample of 6,249 US IPOs that took place between 1980 and 2001.

There is a long history of research devoted to this issue. Researchers on financial economics have been trying to explain the phenomenon decades and it has been the subject of a vast amount of academic work. Many theories have been postulated to offer explanations and it is still a very active field. The mentioned paper by Ritter and Welch is a good introduction to the domain.

Depending on the point of view, underpricing could be seen as money left on the table or a considerable opportunity for those who can get shares allocated on the right IPOs. Given the sums involved, the ability to predict the initial return could be very profitable. Sellers could adjust the offering price to get more money for their shares, and buyers could improve their allocation of funds. Having said that, IPOs underpricing prediction poses a major challenge. The nature of the underpricing is not well understood, the set of variables identified as relevant in the literature is incomplete, and the presence of outliers adds difficulty to the task. In this context we suggest that soft computing and, in particular, evolutionary computation could be very helpful.

The use of evolutionary computation in financial prediction is hardly new. Many pieces of research have explored the suitability of this technique in areas such as trading (Allen and Karjalainen, 1999; Ghandar et al 2009), portfolio optimization (Korczak 2002; Lin and Liu, 2008, Chen et al. 2011) or bankruptcy prediction (Shin, 2002, Gaspar-Cunha et al. 2010). However little has been done in IPO research. Most of the empirical analysis that has been carried out to date in order to explain underpricing through variables related to the offering or the financial structure of the companies is based on linear models. Besides the seminal work of Jain and Nag (1995), Robertson et al. (1998) or Chou et al. (2010), very few efforts have been done using artificial intelligence.

All the mentioned authors tackle the problem in a very similar way. They model it as a regression task and rely on multilayer perceptrons to solve it. Robertson et al. add factorial analysis to understand the effects of different parameter settings and Chou et al. use a genetic algorithm to optimize the architecture of the neural network but, apart from that, there are no major differences. The approach suggested in this paper is very different. We introduce a system that evolves a set prediction rules using a steady state genetic algorithm with a Michigan approach. Once the system is trained, it can subsequently be used to predict underpricing for new IPOs matching relevant rules to the values of the independent variables. One of the main advantages of this system is the way it handles outliers, which are a major factor in this domain.

The rest of the paper will have the following structure: In Section 2, we introduce the explanatory variables and describe the data. Section 3 deals with the methodology and Section 4 will be used to report the results of our empirical analysis. Finally, Section 5 covers the summary and conclusions.

## 2. VARIABLES AND DATA

As a starting point, we provide a formal definition of the phenomenon we have been referring to as IPO underpricing.

We define IPO underpricing as the percentage change of the share price from the offer to the closing price on first day of trading minus the return on the appropriate index or:

$$R_i = \left( \frac{Pc_i - Po_i}{Po_i} \right) - \left( \frac{Mc_i - Mo_i}{Mo_i} \right)$$

Where $R_i$ is the adjusted first day return for stock i; $PO_i$ is the offering price for stock i; $Pc_i$ is the closing price for stock i; $Mo_i$ is the closing for the broad market index of the market where the stock i was floated for the day before the IPO and $M_{Ci}$ is the closing for the broad market index of the market where the stock i was floated on the day of the IPO.

Once the target variable has been clearly described, we introduce the independent variables.

### 2.1. Variables

As we have already mentioned in the introduction, the amount of literature regarding IPO underpricing is quite remarkable. This fact makes the initial number of potential explanatory variables very high. However, there seems to be a number of them concerning the structure of the offerings that show up very often.

These variables, which are about to be succinctly described, are the following: Price, range width, price adjustment, offer price, retained stock offer size and relation to tech sector.

*Price range width (RANGE)*

This variable represents the width of the non-binding reference price range offered to potential customers during the roadshow. This width can be interpreted as a sign of uncertainty regarding the real value of the company and a therefore, as a factor that could influence the initial return. Following Hanley (1993), and Kirkulak and Davis (2005), the representation to be used will be the difference between the maximum and minimum price divided by the minimum price.

*Price adjustment (P_ADJ)*

Hanley (1993) or Ljungqvist and Wilhelm (2003) suggest the relation between the final offer price and the mentioned price range might also be interpreted as sign of uncertainty. They state that this effect might be captured by the following expression.

$$P\_ADJ = \frac{\left|P_f - P_e\right|}{P_e}$$

Where $P_f$ is the final offer price and $P_e$ is the expected price defined as the middle point of the price range.

*Offering price (PRICE)*

The final offering price has been found to be a relevant variable not only as a part of the previous indicator, but on its own. Studies like Chalk and Peavy (1987) or Albring et al. (2007), among others, support this idea.

*Retained stock (RETAINED)*

The influence of the capital retained by initial investors at the time of the IPO has been traditionally understood to have a signal the quality of the stock (Leland and Pyle, 1977; Aggarwal et al., 2002; Ljungqvist and Wilhelm, 2003). Since we lack the breakdown of primary and secondary shares, we will proxy this variable through the ratio number of shares sold at the IPO divided by post-offering number of shares minus the number of shares sold at the IPO.

*Offering size (LSIZE)*

This variable is defined as the logarithm of the offering size in millions of dollars excluding the over-allotment option Studies like Megginson and Weiss (1991), Jain and Kini (1999) or Albring et al (2007) support the need to include it in the models.

*Technology (TECH)*

The reason why we suggest a specific variable to control whether the industrial activities of a company are related to tech sector is the fact that they tend to show a higher underpricing. This fact is usually modelled by a dummy that equals one for tech (Smart and Zutter, 2003; Loughran and Ritter, 2004; Lowry et al. 2010). Our labelling criterion is based on IPO Monitor's definition. This company publishes reports with the list of tech companies taken public based on US Standard Industry Codes. Hence, we will consider an IPO to be "Tech" if it is in the list.

### 2.2 Data

The sample consists of 866 companies taken public between January 1999 and May 2010 in US stock markets. This includes AMEX, NASDAQ and NYSE IPOs and excludes American Depositary Receipts; closed-end funds; real estate investment trusts and unit offerings. Our primary data source was IPO Monitor. The information was completed with IPO profiles from Hoovers. Index information was obtained from NASD (NASDAQ and AMEX composites) and DataStream (S&P 500).

The main descriptive statistics (mean, median, maximum, minimum and standard deviation) for the sample of 866 IPOs considered in the analysis are reported in table 1.

*Table 1:* Descriptive statistics.

| | *Mean* | *Median* | *Max* | *Min* | *Std. Dev.* |
|---|---|---|---|---|---|
| RETURN | 0.176 | 0.073 | 3.718 | -0.281 | 0.389 |
| PRICE | 14.641 | 14.000 | 85.000 | 3.250 | 5.812 |
| RANGE | 0.149 | 0.143 | 0.500 | 0.000 | 0.063 |
| P_ADJ | 0.105 | 0.082 | 1.509 | 0.000 | 0.099 |
| LSIZE | 2.054 | 2.025 | 3.939 | 0.061 | 0.446 |
| RETAINED | 0.309 | 0.262 | 1.000 | 0.002 | 0.198 |
| TECH | 0.328 | 0.000 | 1.000 | 0.000 | 0.470 |

## 3. RULE-BASED SYSTEM

In this section we introduce the rule-based prediction system. The solution that we suggest is based on genetic algorithms (Holland, 1975; Fogel 1994) and inspired by Packard's (1990) prediction method for dynamical systems.

As we mentioned before, the use of genetic algorithms in IPOs faces a major difficulty, namely, the presence of a number of outliers that are hard to characterize. We consider that this factor justifies the choice of a Michigan approach (Booker et al., 1989) where each individual represents the solution to a part of the domain and the solution to the problem is represented by the population as a whole.

In this case, the evolution results in each individual being specialized in a section of the input space and, therefore, providing forecasts only for those patterns within its range of capability. Since the population as a whole is used like a set of rules to predict most of the data set, it is possible to evolve individuals that are specialized in those unusual cases, while others evolve focused on the bulk of the data with a regular behaviour.

We will start providing a general description of the genetic representation, followed by details regarding the fitness function and the genetic operators. Finally, we will describe the creation of the initial population and the prediction process.

### 3.1 Representation

The system generates rules defined in the input variable space in order to make prediction in the output variable space. In this work we use the previously defined set of independent variables (LSIZE, P_ADJ, PRICE, RANGE, RETAINED and TECH) and RETURN as target variable. Every rule defines a range for each of these variables setting an upper bound and a lower bound encoded as a real number. The target variable is special as this range is defined by a predicted output and an expected error. That is, if we have six input variables plus one output, each rule will be represented by an array of 12 real numbers for the bounds plus 2 extra real numbers for prediction and error. These rules represent asserts such as

"*If variable 2 is smaller than 1 and bigger than 0.6, variable 3 is smaller than 0.7 and bigger than 0.4, variable 4 is smaller than 0.6 and bigger than 0.2, and variable 5 is smaller than 0.4 and bigger than 0.1 and variable 6 is smaller than 1 and bigger than 0.5, then the prediction (for the output variable) will be 0.5, with an expected error of 0.1*", expressed as

IF (0.6<v2<1) AND (0.4<v3<0.7) AND (0.2<v4<0.6) AND (0.1<v5<0.4) AND (0.5<v6<1)
THEN prediction = 0.5 +/- 0.1

In this case, the individual would be encoded as follows:

(dc,dc,0.6,1,0.4,0.7,0.2,0.6,0.1,0.4,0.5,1,0.5,0.1)

Where “dc” means “don't care”, as the range for this variable was not explicitly defined. In this case, the rule would cover the whole range of values for the variable, usually standardized within [0,1]. Figure 1 illustrates the individual.

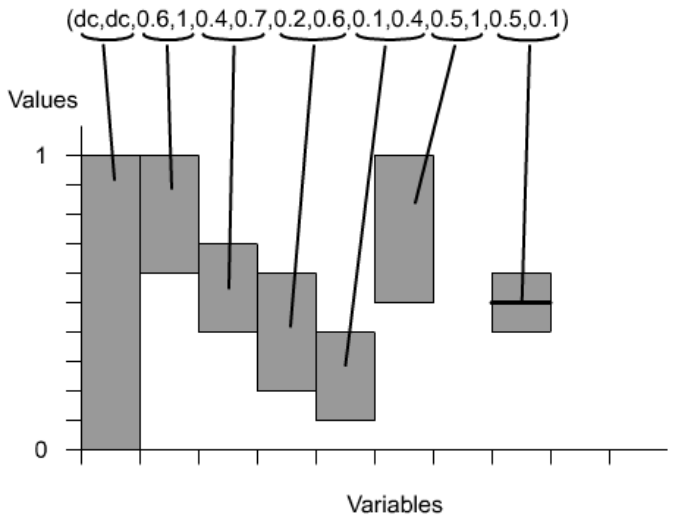


**Figure 1**. Graphical representation of an individual

### 3.2 Fitness computation

The computation of fitness follows the idea suggested by Packard [28], by which good individuals are characterized by uniform values for the dependant variable across matching patterns. The assessment of these individuals (rules), does not take into account the forecasts, which are made at a later stage. Any individual has two associated parameters, a prediction and an error measure. Let *C* be an individual and *i* a training pattern. If the value of the variables for pattern *i* meet the requirements of *C*, we will say *C(i)=true*, otherwise *C(i)=false*. The prediction assigned to the rule will be derived by means of a linear regression fitted to the patterns from the training sample that fulfil the requirements specified by *C*. The error magnitude, *E*, is the maximum absolute difference between the predicted value, $p_i$, and the real value, $v_i$, of the matching training patterns that meet rule *C*.

$$E = \text{Max } i \{|p_i - v_i| / C(i)\}$$

The model favours individuals with the ability to predict the highest number of patterns with the lowest possible error. The specific fitness function that we use is:

$$\text{IF } ((N_C > 1) \text{ AND } (e < E_{MAX})) \text{ THEN } fitness = (N_C * E_{MAX}) - C$$
$$\text{ELSE } \quad fitness = f_{min}$$

where *C* is the individual, and $N_C$ is the number of training patterns that match *C* (in other words, $N_C = \#\{i / C(i)\}$). $E_{MAX}$ is a constant for punishing individuals with a variance greater than its own value, and $f_{min}$ is the lowest value assigned to the individual when the rule is not fitted.

### 3.3 Genetic operators

This subsection describes the genetic operators used. We start defining crossover, followed by mutation and selection and replacement.

*Crossover*

The genetic algorithm paradigm might be useful to evolve new rules such as the one we just mentioned. Two individuals can generate an offspring. This offspring inherits each gene from one parent, as shown in figure 2. A gene is a pair (LL,UL), where LL is the lower limit for a variable, and UL is the upper limit for the same variable. In other words, the offspring receives a gene from each parent with equal probability for each variable. The offspring doesn't inherit parent's predictions and errors. Following the example above:

Parent 1: (0.13, 0.9, 0.2, 0.81, dc, dc, 0.01, 0.94, dc, dc, 0.45, 9.9, 0.73, 0.03)
Parent 2: **(0.14, 0.77, 0.13, 0.85, 0.11, 0.22, dc, dc, 0.32, 0.51, 0.14, 0.27, 0.57, 0.06)**
Offspring: (0.13, 0.9, **0.13, 0.85**, dc, dc, **dc, dc, 0.32, 0.51**, 0.45, 9.9, ? , ?)

The offspring's “prediction” and “error” are not assigned (and therefore are represented as “?”).

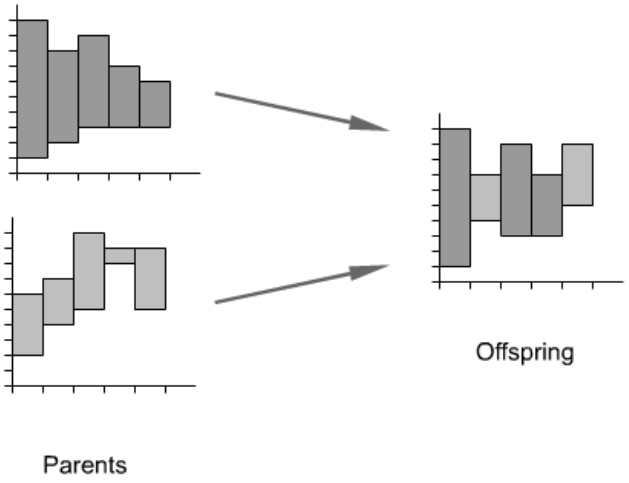


**Figure 2**. Crossover process

*Mutation*

Once generated, an offspring may be subject to a gene mutation. This event takes place with a 10% probability for each new individual. Should the individual be selected, one of its genes will be altered. In table 2, the mutations over the gene (LL,UL) are showed. R(x,y) means a random value between *x* and *y* with $x < y$, and W means the 10% of the width of the interval defined by (LL,UL), in other words, W=0.1(UL-LL). Obviously, the transformation of (LL,UL) into (LL’,UL’) must meet the condition (LL’ < UL’). Figure 3 summarizes these alternatives.

*Table 2:* Mutations.

| *Name* | *Transformation* |
|---|---|
| New random value | (LL,UL) → ( R(0,1), R(0,1) ) |
| Null condition | (LL,UL) → ( dc, dc ) |
| Enlarge | (LL,UL) → (LL-R(0,1)W, UL+R(0,1)W ) |
| Shrink | (LL,UL) → (LL+R(0,1)W, UL-R(0,1)W ) |
| Move up | (LL,UL) → (LL+cW, UL+cW ) <br> c = R(0,1) |
| Move down | (LL,UL) → (LL-cW, UL-cW ) <br> c = R(0,1) |

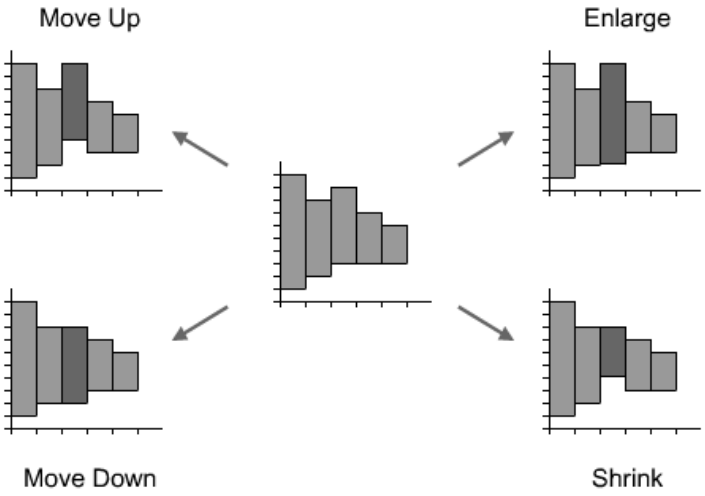


**Figure 3.** Mutation process

*Selection and replacement*

We apply Michigan's approach selecting only two parents by three rounds trial each generation to produce a single offspring, following a Steady-State strategy. This is subsequently used to replace the closest individual in terms of phenotypic distance. That is, we find the individual whose prediction is nearest to the offspring's prediction, and replace it by the offspring if and only if the offspring fitness is better than the individual's fitness. If this doesn't happen, there isn't any change this generation. That strategy generates a diverse population, in which each individual makes a prediction that is different to the rest instead of genetic clones of the best individual, produced by the standard genetic algorithm method.

### 3.4 Initial population and prediction

The generation of the initial population is generated dividing the prediction range (i.e., the interval [0,1]) in 100 subintervals that are 0.01 wide. Then, we create an individual for each prediction interval. For

example, for interval [0.34, 0.35] we search for all the patterns *i* in the training set such that the predicting value *p(i)* is in the [0.34,0.35] interval, and we create an individual with an upper and lower limit for every input variable. For each input variable $x_j$ the upper limit must be the minimum number *U* such that for each pattern i with predicting value *p(i)* in the [0.34,0.35] interval, the value of $x_j$ is less than *U*. Accordingly, we select the lower limit for $x_j$ as the maximum *L* such that for each pattern i with predicting value *p(i)* in the [0.34,0.35] interval, the value of $x_j$ is greater than *L*. We repeat the process for each variable and interval to generate the initial population of 100 individuals and so we can assure that there is an individual in each interval of prediction.

The prediction process relies of the rules generated by the described algorithm. After each execution of the model (500.000 generations), we store in a file, that we call “rule pool”. This pool consists of individuals that match more than 3 patterns in the training set. This process is repeated 10 times and a file with a set of individuals is generated. The prediction for a test pattern is provided by the matching individual. In case more than one rule is relevant for the same pattern of the test set, the final prediction is the median of all the individual predictions. In case there are no matching rules, the system does not provide an estimate.

## 4. EXPERIMENTAL ANALYSIS

The aim of this paper is introducing a GA-based tool that useful for IPO underpricing prediction. We tested the system comparing its forecasting ability to the one achieved by a set of widely used machine learning algorithms. This experimentation extended the results reported in a conference version of this work (Quintana et al., 2005) in several ways. We used recent data, benchmarked the results against a comprehensive set of machine learning algorithms instead of linear models, and the results underwent more rigorous statistical testing.

The algorithms were tested using a 100-fold cross-validation procedure on the data set introduced in section 2. The sample included 866 patterns, each representing a sigle IPO, that was split in two in every fold. The first 780 patterns were used for training, and the next 86 for validation purposes. We applied all the algorithms to the same data sets in order to make the results comparable.

Regarding the experimental parameters, we tested 10, 50 and 100 individuals with an EMAX of 0.55. Given the stochastic nature of genetic algorithms, all the experiments were repeated 10 times. The accuracy of the predictions was assessed comparing the root mean square error (RMSE) of the predictions of the models. RMSE is formally defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (t_i - y_i)^2}{n}}$$

Where $t_i$ is the observed underpricing for IPO *i*; $y_i$ is the prediction for IPO *i* underpricing; and *n* is the number of predicted patterns.

The mean results of such experiments are reported on Table 3. As me mentioned in section 3, the system has the feature of anticipating which patterns could lead to potentially inaccurate forecasts. In these cases, the system does not provide any estimate. This feature is especially important in this

domain, where outliers are frequent. The last column, “Percent of prediction” shows the mean percentage of points in the test data set for which the system could make at least one prediction and, therefore, provided an estimate for underpricing. As we can see, there is an inverse relationship between the number of individuals and both the prediction error and the number of patterns for which there is a prediction available. The system identifies potentially unreliable predictions for 1.1% to 4.6% of the sample.

*Table 3*: Cross-validation for the Rule System. Average root mean squared prediction error, variance of RMSE and percentage of test patterns predicted over 10 repetitions of a 100-fold cross validation

| *Individuals* | *RMSE* | *Var. RMSE* | *Percent of prediction* |
|---|---|---|---|
| 10 | 0.0796 | *0.0636* | 98.94% |
| 50 | 0.0667 | *0.0482* | 96.7% |
| 100 | 0.0650 | *0.0468* | 95.4% |

As we mentioned before, we repeated the same cross-validation analysis using a set of machine learning alternatives. For such task, we used a well-known and widely used Java package called WEKA (Waikato Environment for Knowledge Analysis) (Hall et al., 2009). The algorithms we used as benchmarks are conjunctive rules, IBK, LWL, K*, multilayer perceptrons, M5 Rules, M5P, radial basis neural networks and SMO-Regression.

- Conjunctive Rules: rule system based on information gain.

- IBK (Aha and Kibler, 1991): based on the K-nearest neighbour classifier algorithm.

- LWL (Atkeson et al., 1997): local instance-based weighted learning algorithm. It builds a classifier from the weighted instances.

- K* (Cleary and Trigg, 1995): instance-based classifier, that is the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function.

- M5Rules (Hall et al., 1999): generates a decision list for regression problems using separate-and-conquer.

- M5P (Quinlan, 1992): numerical classifier that combines a conventional decision tree with the possibility of linear regression to predict continual variables.

- Multilayer Perceptron: Algorithm that simulates the biological process of learning through weight adjusting using backpropagation algorithm (Rumelhart et al., 1986).

- RBNN (Moody and Darken, 1989): Radial Basis Neural Networks are another type of artificial neural network. It uses radial basis functions to approximate different regions of the input space depending on their characteristics.

- SMO-Reg (Smola and Scholkopf, 1998): Implements sequential minimal optimization algorithm for training a support vector regression using polynomial or RBF kernels.

The algorithms were tested on the same 100 sets used by the rule system and, once again, the stochastic elements of some of the algorithms called for repeated execution. The experiments were repeated 10 times. The mean error of such experiments is reported on table 4.

*Table 4:* Average root mean squared prediction error of variance of RMSE over 10 repetitions of a 100-fold cross validation for a set of 9 machine learning algorithms

| *Algorithm* | *RMSE* | *Var. RMSE* |
|---|---|---|
| Conj. Rule | 0.0849 | 0.0605 |
| IBK | 0.0767 | 0.0602 |
| K* | 0.0780 | 0.0601 |
| LWL | 0.0834 | 0.0599 |
| M5P | 0.0830 | 0.0661 |
| M5 Rules | 0.0850 | 0.0677 |
| MLP | 0.0948 | 0.0767 |
| RBNN | 0.0813 | 0.0586 |
| SMO-Reg | 0.0766 | 0.0511 |

The statistical significance of the observed differences was formally assessed and the results of these tests for the scenario with the lowest pattern requirements (highest errors and highest prediction rates) are reported in table 5. There, the elements of the rows are compared with the ones of the columns. The sign ”- -” represents that the element in the row is significantly smaller than the one in the column at 1%. The opposite relation is represented by ”++”. Finally,”=” means that there is no significant difference between them. The data in the table was derived using the protocol that follows. First we, tested for normality using Kolmogorov-Smirnov, if both distributions were normal, we analyzed the variances using Levene’s test. Depending on these results, we subsequently performed a t-test or a Welch test. In case on non-normality, we used Wilcoxon test.

*Table 5:* Significance tests for the average prediction error

| | *RS 10* | *RS 50* | *RS 100* | *CR* | *IBK* | *K** | *LWL* | *M5P* | *M5R* | *MLP* | *RBNN* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *RS 50* | -- | | | | | | | | | | |
| *RS 100* | -- | = | | | | | | | | | |
| *CR* | = | ++ | ++ | | | | | | | | |
| *IBK* | = | ++ | ++ | = | | | | | | | |
| *K** | = | ++ | ++ | = | = | | | | | | |
| *LWL* | = | ++ | ++ | = | = | = | | | | | |
| *M5P* | = | ++ | ++ | = | + | = | = | | | | |
| *M5R* | = | ++ | ++ | = | ++ | + | = | = | | | |
| *MLP* | ++ | ++ | ++ | ++ | ++ | ++ | ++ | + | + | | |
| *RBNN* | = | ++ | ++ | = | = | = | = | = | - | -- | |
| *SMO* | -- | + | + | -- | -- | -- | -- | -- | -- | -- | -- |

The best alternative to the rule system seems to be SMO-Reg, closely followed by IBK and K*. The support vector machine offers both the lowest mean RMSE and variance among its peers. If we

compare these results to the ones provided by the rule system we see that the latter offers the lowest RMSE for 50 and 100 individuals. For 10, SMO is significantly more accurate at 1% but, apart from that case, the rule system consistently beats the alternatives. If we compare the tree configurations of the rule system among themselves, find that the prediction error for RS 10 is significantly higher than the other two. RMSE for RS 100 seems lower than for RS 50, but is identical in statistical terms at 1%.

These results show that the rule system based of genetic algorithms is competitive and, therefore, could potentially be a useful tool in this domain.

## 5.SUMMARY AND CONCLUSSIONS

In this paper we have discussed an implementation of rule system based on genetic algorithms to predict the underpricing of IPOs. The predictions are based on six variables identified by literature review and their accuracy has been compared to the forecasts provided by several machine learning algorithms.

The rule system introduced is based on a genetic algorithm with a Michigan approach that provides predictions based on a set of matching rules. The system evolves specialised rules to be used with patterns that show similar behaviour. This way, the population encodes rules specialised on different sections of the space instead of general ones. This feature is especially relevant in domains like the one we are dealing with, where the presence of outliers makes prediction particularly difficult.

The algorithm was tested on a US sample of IPOs using repeatedly a 100-fold cross validation analysis for three configurations. The system offers results that are very competitive. Only a support vector machine offered a lower prediction error for one configuration. For the other two RMSE was significantly better in every case. The system identified number of patterns for which predictions could potentially be unreliable that ranged between 1.1% and 4.6% of the sample.

These results are encouraging and suggest that IPO research would benefit from the use of this tool. We understand that so would any prediction effort in domains where the identification of outliers is particularly challenging.

## 6. ACKNOWLEDGMENTS

We are grateful to two anonymous referees for valuable suggestions and comments. We also acknowledge financial support granted by the Spanish Ministry of Science under contract TIN2008-06491-C04-03 (MSTAR) and Comunidad de Madrid (CCG10-UC3M/TIC-5029).

## 7. REFERENCES

Aggarwal, R.K., Krigman L., Womack K. 2002, Strategic IPO underpricing, information momentum, and lockup expiration selling. *Journal of Financial Economics* **66**, 105-137.

Aha, D., Kibler D., 1991, Instance-based learning algorithms. *Machine Learning* **6**, 37-66.

Albring, S.M., Elder, R.J., Zhou, J., 2007, IPO Underpricing and Audit Quality Differentiation within Non-Big 5 Firms. *International Journal of Auditing* **11**, 115–131.

Allen F., Karjalainen R., 1999, Using genetic algorithms to find technical trading rules. *Journal of Financial Economics* **51**, 245-272.

Atkeson, C.G., Moore, A.W., Schaal, S., 1997, Locally weighted learning. *Artificial Intelligence Review* **11**, 11-73.

Balvers, R.J., McDonald B., Miller R.E., 1988, Underpricing of new issues and the choice of auditor as a signal of investment banker reputation. *Accounting Review* **63**, 605-622.

Booker, L.B., Goldberg, D.E., Holland, J.H., 1989, Classifier systems and genetic algorithms. *Artificial Intelligence* **40**, 235-282.

Chalk A., Peavy, J., 1987, Initial public offerings, daily returns, offering types and the price effect. *Financial Analysts Journal* **43**, 65-69.

Chen Y., Mabu, S., Hirasawa, K., 2011, Genetic relation algorithm with guided mutation for the large-scale portfolio optimization. *Expert Systems with Applications* **38**, 3353-3363.

Chou, S., Yen-Sen, N., William T.L., 2010, Forecasting IPO price using GA and ANN simulation. In *Proceedings of the 10th WSEAS international conference on signal processing, computational geometry and artificial vision (ISCGAV'10)*, Taipei, Taiwan,145–150.

Cleary, J.G., Trigg, L.E., 1995, K*: An instance- based learner using an entropic distance measure. *Proceedings of the 12th International Conference on Machine learning*, Tahoe City, USA, 108-114.

Fogel, D.B., 1994, An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks* **5**, 3-14,

Gaspar-Cunha, A., Mendes, F., Duarte, J., Vieira, A., Ribeiro, B., Ribeiro, A., Neves, J., 2010, Feature selection for bankruptcy prediction: a multi-objective optimization approach. *International Journal of Natural Computing Research* **1**, 71-91.

Ghandar, A., Michalewicz, Z., Schmidt, M., To, T.D., Zurbrugg R., 2009, Computational intelligence for evolving trading rules. *IEEE Transactions on Evolutionary Computation* **13**, 71-86.

Hall, H., Holmes, G., Frank E., 1999, Generating rule sets from model trees. *Proceedings of the Twelfth Australian Joint Conference on Artificial Intelligence*, Sydney, Australia,1-12.

Hall, M., Frank E., Holmes G., Pfahringer B., Reutemann, P., Witten I., 2009, The WEKA data mining software: an update. *SIGKDD Explorations* **11**, 10-18.

Hanley, K.W., 1993, The underpricing of initial public offerings and the partial adjustment phenomenon. *Journal of Financial Economics* **34**, 231-250.

Holland, J.H., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

Jain, B.A., Kini, O., 1999, On investment banker monitoring in the new issues market. *Journal of Banking and Finance* **23**, 49-84.

Jain, B.A., Nag, B.N., 1995, Artificial neural network models for pricing initial public offerings. *Decision Sciences* **26**, 283-299.

Kirkulak, B., Davis, C., 2005, Underwriter reputation and underpricing: evidence from the Japanese IPO market. *Pacific-Basin Finance Journal* **13**, 451-470.

Korczak, J., Lipinski, P., Roger, P., 2002, Evolution strategy in portfolio optimization. in Collet, P., Fonlupt, C.,Hao, J.K., Lutton, E., Schoenauer, M. (Eds.) *Artificial Evolution*. Lecture Notes in Computer Science 2310. Springer, Berlin,156-167.

Leland, H., Pyle, D.,1977, Informational asymmetries, financial structure and financial intermediation. *Journal of Finance* **32**, 371-387.

Lin, C.C., Liu, Y.T., 2008, Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research* **185**, 393-404

Loughran, T., Ritter, J.R., 2004, Why has IPO underpricing changed over time? *Financial Management* **33**, 5-37.

Lowry, M., Officer, M.S., Schwert, W., 2010, The variability of IPO initial returns. *Journal of Finance* **65**, 425-465.

Ljungqvist, A., Wilhelm, W., 2003, IPO pricing in the dot-com bubble. *Journal of Finance* **58**, 723–752.

Mann, H.B., Whitney, D.R., 1947, On a Test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics* **18**, 50-60.

Megginson, W.L., Weiss K.A., 1991, Venture capitalist certification in initial public offerings. *Journal of Finance* **46**, 799-903.

Moody, J., Darken, C.J., 1989, Fast learning in networks of locally tuned processing units. *Neural Computation* **1**, 281-294.

Packard, N.H., 1990, A genetic learning algorithm for the analysis of complex data. *Complex Systems* **4**, 543-572.

Platt, J., 1998, Fast training of support vector machines using sequential minimal optimization. in B. Schoelkopf, C. Burges, and A. Smola, (Eds.) *Advances in Kernel Methods - Support Vector Learning*, Cambridge: MIT Press.

Quinlan R.J.,1992, Learning with continuous classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, Singapore, Singapore, 343-348.

Quintana, D., Luque, C., Isasi, P., 2005, Evolutionary rule-based system for IPO underpricing prediction. In *Proceedings of genetic and evolutionary computation conference (GECCO 2005),* Washington D.C., USA, 983–989.

Ritter, J.R., Welch, I., 2002, A review of IPO activity, pricing, and allocations. *Journal of Finance* **57**, 1795-1828.

Robertson, S.J., Golden, B.L., Runger, G.C., Wasil, E.A., 1998, Neural network models for initial public offerings. *Neurocomputing* **18**, 165-182.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986, Learning representations by back-propagating errors. *Nature* **323**, 533-536.

Shin, K., Lee, Y., 2002, A genetic algorithm application in bankruptcy prediction modeling. *International Journal of Expert Systems with Applications* **23**, 321-328.

Smart, S.B., Zutter, C.J., 2003, Control as a motivation for underpricing: a comparison of dual and single-class IPOs. *Journal of Financial Economics* **69**, 85-110.

Smola, A.J., Schölkopf, B., 1998, *A tutorial on support vector regression*. NeuroCOLT2 Technical Report Series - NC2-TR-1998-030.